REMARKS

This is intended as a full and complete response to the Final Office Action dated October 18, 2005, having a shortened statutory period for response set to expire on January 18, 2006. Applicants submit this response to place the application in condition for allowance or in better form for appeal. Please reconsider the claims pending in the application for reasons discussed below.

Claims 1, 3-10, 12-19 and 21-26 are pending in the application. Claims 1, 3-10, 12-19 and 21-26 remain pending following entry of this response.

Claim Rejections - 35 U.S.C. § 102

Claims 1, 3-10, 12-19 and 21-26 are rejected under 35 U.S.C. 102(e) as being anticipated by *Bauman et al.* (U.S. 2003/0097488, hereinafter *Bauman*).

Respectfully, *Bauman* fails to qualify as a reference under 35 § U.S.C. 102(e) against the present Application. Applicants filed the present application on January 4, 2002, with a preliminary amendment. The preliminary amendment includes a priority claim to U.S. 09/990,850 with a U.S. filing date of November 21, 2001. In point of fact, the present rejection claims priority to *Bauman*. The filing receipt received by Applicants reflects the priority claim to *Bauman*. Applicants submit that the rejection of the present Application under 35 § U.S.C. 102(e) using its own parent is improper, and therefore, respectfully request that the rejection be withdrawn.

Claim Rejections - 35 U.S.C. § 103

Claims 1, 4-8, 10, 13-17, 19, 21 and 23-26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Admitted Prior Art (APA) in view of *Firth et al.* (US. 5,987,517, hereinafter *Firth*). Respectfully, applicants traverse the rejection.

The Examiner bears the initial burden of establishing a *prima facie* case of obviousness. *See* MPEP § 2142. To establish a *prima facie* case of obviousness three basic criteria must be met. First, there must be some suggestion or motivation, either in

Page 8

419092_1

the references themselves or in the knowledge generally available to one ordinary skill in the art, to modify the reference or to combine the reference teachings. Second, there must be a reasonable expectation of success. Third, the prior art reference (or references when combined) must teach or suggest all the claim limitations. *See* MPEP § 2143. The present rejection fails to establish at least the first and third criteria.

For example, claim 1 recites the limitation of "in response to a request from the client, issuing a single, continuous mode operation to the socket." Claims 10 and 19 each recite a corresponding limitation. Although the Examiner argues that the APA discloses this limitation at p. 3, lns. 13-16 and p.3 lns. 9-10, these passages are in fact directed to prior art methods that require multiple accept() or receive() operations be performed as part of a socket-based communication exchange. Set out in full, this passage provides:

> Sockets accept connections and receive data from clients using well-known "accept" and "receive" semantics, respectively. The accept and receive semantics are illustrated in FIGS. 1 and 2 as accept ( ) and asyncAccept ( ), respectively, and as receive ( ) and asyncReceive ( ), respectively. Sockets accept/receive semantics are either synchronous (FIG. 1) or asynchronous (FIG. 2). Synchronous accept/receive APIs accept connections and receive data in the execution context issuing the API. Asynchronous APIs such as return indications that the accept/receive will be handled asynchronously if the connection/data is not immediately available.

Application, p.3, 9-16. Nothing in this passage (or the APA generally) discloses a method of socket-based communication that includes issuing a "continuous mode operation to the socket." Rather, the passage cited by the Examiner describes the existence of accept() and receive() calls as known socket-based communication semantics, a point Applicants do not dispute. As Applicants point out in the APA, however, the prior art methods of actually using socket-based accept/receive communications often require the use of multiple, inefficient accept() and receive() calls:

> As suggested by the loops shown in FIG. 1 and FIG. 2, accept and receive processing for both synchronous and asynchronous environments is

Page 9

> highly repetitive with little variance in the parameters. As a result, a server may service thousands of accepts and receives in a short period of time. Because this is such a common path, it would be desirable to eliminate or reduce redundant accept and receive processing.

Application, p. 3, lns. 26-30. Moreover, the Examiner appears to concede this very shortcoming of the prior art in that the "APA does not explicit teach [sic] the single asynchronous operation." *See Final Office Action*, p. 3. In other words, the APA highlights that socket-based communication often require the use of multiple accept() / receive() calls as part of a socket-based communication.

Nevertheless, the Examiner argues that *Firth* renders obvious the techniques for socket-based communications, as recited in the present claims. However, the material cited from *Firth* (and *Firth* generally) fails to disclose techniques for managing socket-based communications. Rather, *Firth* is directed to an API of functions providing an abstraction of data-communication techniques. The abstractions allow developers of high-level applications to create applications without having to understand or manage the details of an underlying communication mechanism. That is, the abstraction provided by the "Internet API" allows developers to create software applications without having to understand how a particular socket-based communication may occur.

> In the preferred embodiment of the present invention, calls to two of the reentrant Internet API functions (e.g., InternetOpen(. . .),InternetConnect(. . . ,FTP, . . .), which will be explained in detail below) will initialized an Internet session, establish a connection, and manage all the underlying details including the FTP protocol, the communication facilities required (e.g., a socket connection), ... **The Internet application program does not have to include source code to establish an Internet connection, handle the FTP protocol, the communications facilities, or the underlying protocols. All of these details are abstracted In the Internet API and are hidden from or transparent to the application program.**

Firth, 3:37-52. As the emphasized passage makes clear, *Firth* discloses techniques that allow an application developer to compose an application without an understanding of "the underlying protocols." Not surprisingly, therefore, *Firth* fails to teach or suggest

Page 10

mechanisms for providing asynchronous network communications, in the manner claimed.

The three passages cited by the Examiner actually serve to highlight that *Firth* is directed to functions provided by an "internet API," and not to techniques for performing socket-based communications, as recited by claims 1, 10, and 19. For example, the first passage form *Firth* provides in part: "As was just described, a single call to the InternetOpen ( ) function from the internet API provides a client application with the ability to select the type of Internet access, select a proxy for a first level of security, ... [or perform other various actions]." *Firth*, 16:23-27. None of the actions provided by the "InternetOpen() API call include "issuing a single, continuous mode operation" to a socket using a "single asynchronous accept operation" or a "single asynchronous receive operation." Rather, the functions of the "internet API" disclosed in *Firth* are provided to operate independently from the mechanism used to manage socket-based communications. In fact, this is the whole point of the "internet API": to provide an API where the details of a socket-based communication are "abstracted in the Internet API and are hidden from or transparent to the application program." *Firth*, 3:50-52. Similarly, the other two passages cited by the Examiner are directed to different aspects of the high-level "Internet API" disclosed by *Firth*.

Finally, the Examiner asserts that "it would have been obvious to one of the ordinary skill in the art at the time the invention was made to combine the teaching of APA and *Firth* because *Firth's* single asynchronous operation would improve flexibility of APA's system by adding new or additional Internet application protocols for establishing communications with a variety of computer networks." *See Final Office Action*, p. 3. However, the Examiner bases this assertion one of the passages cited in the rejection of claims 1, 10, and 19. Specifically, the Examiner paraphrases a passage from *Firth* that describes the use of an "InternetConnect()" function call provided by the "InternetAPI." *Firth* describes how this function call may be used to initiate a connection

Page 11

419092_1

using many different communication protocols, as opposed to having multiple function calls, one for each different protocol:

> [Using the"InternetConnect()" function call] an application can communicate common information about several requests using a single function call. In addition, this single application protocol connection function call provides flexibility for adding new or additional Internet application protocols.

Firth, 18:15-20. In describing the InternetConnect()" function call, Firth goes on to provide: "However, the function call and the number off arguments would remain the same and provide a consistent interface for applications." *Firth*, 18:23-26. Applicants submit that the "InternetAPIs" use of a single "InternetConnect()" function call with multiple different parameters, representing multiple different communication protocols, is unrelated to use of a "a single asynchronous accept operation," and "a single asynchronous receive operation" to reduce the number of accept() or receive() calls made as part of a socket-based communication exchange.

For all the foregoing reasons, Applicants submit that independent claims 1, 10, and 19 are allowable. Therefore, Applicants request withdrawal of this rejection and the allowance of these claims.

Claims 3 and 12 are rejected under 35 U.S.C. 103(a) as being unpatentable over Admitted Prior Art (APA) in view of *Firth*, as applied to claim 1 above, and further in view of *Shah et al.* (U.S. Patent 6,175,879 B1, hereinafter *Shah*).

Applicants submit that because APA in view of *Firth*, fails to teach or suggest the invention claimed in independent claim 1 and 10, for the reasons stated above, the rejection of claims 3 and 12 is obviated without the need for further remarks by Applicant.

Claims 9, 18, and 22 are rejected under 35 U.S.C. 103(a) as being unpatentable over *APA* in view of *Firth*, and further view of *Joh* (U.S. Patent 6,175,879B1 "*Joh*").

Page 12

419092_1

Applicants submit that because APA in view of *Firth*, fails to teach or suggest the invention claimed in independent claim 1 ,10, and 19, for the reasons stated above, the rejection of claims 9, 18, and 22 is obviated without the need for further remarks by Applicant.

Conclusion

Having addressed all issues set out in the office action, Applicants respectfully submit that the claims are in condition for allowance and respectfully request that the claims be allowed.

If the Examiner believes any issues remain that prevent this application from going to Issue, the Examiner is strongly encouraged to contact Gero McClellan, attorney of record, at (336) 643-3065, to discuss strategies for moving prosecution forward toward allowance.

Respectfully submitted,

/Gero G. McClellan, Reg. No. 44,227/
Gero G. McClellan
Registration No. 44,227
PATTERSON & SHERIDAN, L.L.P.
3040 Post Oak Blvd. Suite 1500
Houston, TX 77056
Telephone: (713) 623-4844
Facsimile: (713) 623-4846
Attorney for Applicants

Page 13

419092_1